10-2021

Privacy-preserving voluntary-tallying leader election for Internet of Things

Tong WU

Guomin YANG
Singapore Management University, gmyang@smu.edu.sg

Liehuang ZHU

Yulin WU

Citation

WU, Tong; YANG, Guomin; ZHU, Liehuang; and WU, Yulin. Privacy-preserving voluntary-tallying leader election for Internet of Things. (2021). *Information Sciences*. 574, 461-472.
Available at: https://ink.library.smu.edu.sg/sis_research/7330

Information Sciences 574 (2021) 461–472

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Privacy-preserving voluntary-tallying leader election for internet of things

Tong Wu [a], Guomin Yang [b,*], Liehuang Zhu [a], Yulin Wu [c]

[a] School of Cyberspace Science and Technology, Beijing Institute of Technology, China
[b] Institute of Cybersecurity and Cryptology, School of Computing and Information Technology, University of Wollongong, Australia
[c] School of Computer Science and Technology, Harbin Institute of Technology, China

ARTICLE INFO

Article history:
Received 13 August 2020
Received in revised form 9 May 2021
Accepted 9 June 2021
Available online 12 June 2021

Keywords:
IoT
Blockchain
Voluntary-tallying
Voter anonymity
Ballot privacy

ABSTRACT

The Internet of Things (IoT) is commonly deployed with devices of limited power and computation capability. A centralized IoT architecture provides a simplified management for IoT system but brings redundancy by the unnecessary data traffic with a data center. A decentralized IoT reduces the cost on data traffic and is resilient to the single-point-of-failure. The blockchain technique has attracted a large amount of research, which is redeemed as a perspective of decentralized IoT system infrastructure. It also brings new privacy challenges for that the blockchain is a public ledger of all digital events executed and shared among all participants.

The decentralized IoT system relies on the leader election deeply to implement the decentralized communications among the distributed nodes. The conventional leader election must have a centralized authority, contrasting to the decentralization. As an alternative, self-tallying type schemes have been proposed in the literature for decentralized systems. These schemes suffer from adaptive and abortive issues. Also, some additional factors should be considered, such as the availability of candidate nodes. If the candidate node is unavailable after the voting phase due to being offline or ongoing tasks, the next available candidate should be elected. To accommodate such a need, in this paper, we propose a new leader election paradigm called voluntary-tallying leader election, which achieves the core requirements such as ballet secrecy, voter privacy and the additional feature of voluntary-tallying. We formalize the system and security models for this new election paradigm and present a secure and practical construction.

1. Introduction

The continuous developing wireless and mobile technologies have enabled the communication among a massive number of devices such as computers, smartphones, and daily-life devices (e.g. refrigerators, vehicles and wearable devices), which brings the blossom of the Internet-of-Things (IoT) [1–5]. These devices are allowed to communicate with other devices without any human interaction. However, the traditional centralized IoT infrastructure brings some disadvantages, such as communication and computation redundancy, single point of failure, surveillance, easy target for cyber-criminals and

[*] Corresponding author.
E-mail address: gyang@uow.edu.au (G. Yang).

https://doi.org/10.1016/j.ins.2021.06.028
0020-0255/© 2021 Elsevier Inc. All rights reserved.

proprietary solutions without external security verification [6]. Fortunately, the above problems can be addressed by the decentralized IoT infrastructure.

Some significant blockchain-based works in IoT systems have been proposed aiming to address the vehicle management system [7], edge computing [8], and data management [9] by access control mechanism [10]. In the aforementioned works, the leader election plays an essential role. General speaking, the leader election is to select a special node as the manager or gateway for coordinating a group of energy, communication, and computation constrained IoT nodes in various tasks. In practice, the leader node is chosen by some common principles such as their capability, proximity to base station, the ratio between its residual energy and the average energy of the cluster, and the sum of distances between the node and others nodes [11]. Leader election is crucial in some critical IoT systems, e.g., the healthcare systems like the AlarmNet [12], which consists of devices monitoring heart rate, pulse rate, temperature, etc., for the patients. The leader node will serve as a communication gateway between the IoT devices and the server (e.g., the IoT cloud), which is illustrated in Fig. 1. It can also serve as a coordinator for the IoT devices in certain tasks, e.g., software driver update for a group of devices. Therefore, it is important to ensure that the leader node is elected fairly by all the legitimate nodes to avoid the leader node from being controlled by an attacker.

Leader election has been extensively studied in the literature. The conventional leader election schemes are based on some privacy-preserving cryptographic tools and a trusted authority is required to compute the tally, include: the mix-based [13], homomorphic encryption-based [14], and blind signature-based [15].

To address the obstacle in the decentralized setting, the self-tallying type schemes [16–20,1] were proposed to allow the tally to be performed by the voters themselves. However, the self-tallying type schemes are inherently vulnerable to the adaptive and abortive attacks performed by some voters. The abortive issue indicates that some of the voters refuse to vote and give up before casting their ballots. The adaptive issue indicates that the last voter has the priority to access the final result. Thus, it may affect the choice of the last voter an even cause the last voter to give up. Solving adaptive and abortive issues is a challenge to the existing self-tallying type schemes.

Besides the adaptive and abortive issues in self-tallying leader election, we observe that there is another important feature that is desirable for leader election in IoT but has been neglected in the literature. Due to the dynamic state of an IoT device and its constrained resources and computation power, it is possible that an leader node elected by other IoT nodes is not the best candidate, e.g., due to its current system state, availability or workload. Therefore, if a winning candidate is unavailable, e.g., being offline or overloaded by other tasks, it is desirable to allow the next available candidate to be elected.

1.1. Contribution

To achieve the aforementioned features, in this paper, we propose a new election paradigm called voluntary-tallying leader election. In a nutshell, besides the common features required by an election scheme, a voluntary-tallying leader election

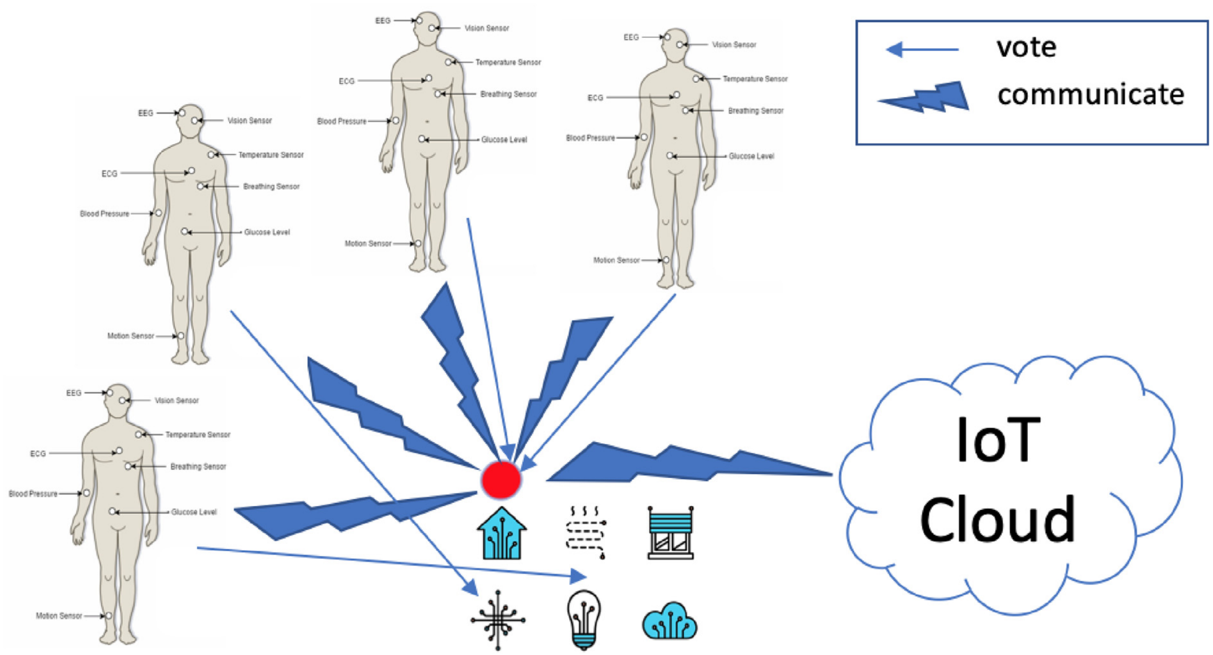


Fig. 1. Leader election in healthcare systems: the sensors in the body-area network select the leader node as the preprocessor of the sensors' data and the gateway to the IoT cloud.

requires that a candidate is voluntary to claim its ballots in the tallying phase. Specifically, a voluntary-tallying leader election should achieve the following features:

- Decentralization: there is no central authority involved in the whole leader election process.
- Ballot Secrecy: the partial tally corresponding to the ballots of a set of voters (beyond what is known and computed trivially by merely having the final tally) is not accessible to any other voters in any phase.
- Soundness: each legitimate voter can only cast one ballot, and the ballot can only be claimed by the intended candidate in the tallying phase. Specifically, a candidate cannot claim a ballot that belongs to other candidates.
- Fairness: no information of the tally is known to any voter, including the last voter, in the voting phase. It means that none of the voting nodes is able to take adaptive or abortive actions before they cast the ballot.
- Voluntary-tallying: in the tallying phase, a candidate can voluntarily claim ballots that voters cast to him/her. In particular, if a candidate is unavailable due to some reason, he/she can choose not to claim the ballots. Then, the next available candidate with the highest number of votes will be elected.
- Dispute Freeness: it ensures that any party can verify that all the claimed ballots are counted towards the correct candidate in the tallying phase.
- Voter Anonymity: it ensures that voters' identities are not disclosed during the entire process.

We formalize the system and security models for the proposed voluntary-tallying leader election framework and present a proven secure construction. We also implement our construction to evaluate its performance. The performance analysis indicates that the instantiation is efficient and practical.

1.2. Organization

The rest of the paper is organized as follows: We introduce the related works including the blockchain-based IoT applications and cryptographic technique on leader election in Section 2. We introduce some preliminary, primitives, system model and the security models in Section 3. We construct a blockchain-based voluntary-tallying leader election in Section 5 then prove the security of our construction in Section 6, respectively. The implementation of the proposed protocol is illustrated in Section 7. Finally, we conclude our work in Section 8.

2. Related work

The blockchain-based applications for IoT systems have attracted a large amount of attention. The blockchain-based vehicle management system [7], blockchain-based edge computing [8], and blockchain-based data management system [9] are all mentioned that the communication to the internet must be with the leader node. Considering the security and privacy issues in IoT system, the leader election in IoT systems commonly adopts the cryptography-based voting schemes.

In conventional centralized voting schemes, a central trusted authority is deployed for organizing the voting and tallying including: the mix-based [13,21], homomorphic encryption-based [14], and blind signature-based [15]. Motivated by removing the trusted third party, Kiayias and Yung [16] suggested a voting scheme, characterized by " self-tallying". In a self-tallying voting scheme, anyone can check the ballot validity and compute the valid ballots to get the voting result. The concrete construction achieves perfect ballot privacy and dispute-freeness. It ensures the partial tally of the ballots of any set of voters that is able to be computed only by the coalition of all the remaining voters and the integrity of the vote. Groth et al. [22] proposed a voting scheme with better efficiency, which adopts the anonymous broadcast channel to achieve the perfect message secrecy. Hao et al. [18] proposed a lower round complexity self-tallying voting scheme for large-scale voting, which is based on a two-round anonymous veto protocol (AV-net) [17]. However the aforementioned works are vulnerable to the adaptive and abortive issues. If voters refuse to vote before casting their ballots or the last voter accesses the final result prior to others, it may affect the choice of the last voter and even cause the last voter to give up.

Considering the development of blockchain technique in voting schemes, some existing voting schemes make use of the blockchain as a public bulletin board and provide the voter privacy with a trusted authority, such as FollowMyVote [23] and TIVI [24]. Unlike theses works, [19] is the first implementation of a decentralized self-tallying voting scheme [18] which is based on blockchain to compute the tally result and to protect the voter's privacy. Additionally, [1] introduces a framework of self-tallying systems in decentralized IoT based on blockchain. Their work satisfies the Maximum Ballot Privacy (MBP) suggested in [18] and partially solves the adaptive and abortive issues. In [20], the authors proposed the first decentralized ranked-choice online voting system implemented in Ethereum's Solidity language. To provide transparency, integrity, and confidentiality for reliable online voting, they suggested to combine the blockchain technology and modern cryptography. They implemented a decentralized online voting system as a smart contract on the Ethereum blockchain. It eliminates hardwired restrictions on possible vote assignments to candidates by using homomorphic encryption to protect voter confidentiality and storing proof of knowledge for each element of a vote. se Table 1.

However, choosing a leader node in IoT system is not a simple case that the system chooses the node with the highest votes, but the system has to balance the workload and the state of node, simultaneously. The aforementioned works do

Table 1
The State of Art.

	Type	Ballot Privacy	Voter Privacy	Abortive	Adaptive
[14]	Centralized	√	×	NA	NA
[15]	Centralized	√	×	NA	NA
[21]	Centralized	√	×	NA	NA
[16]	Self-Tallying	√	×	×	×
[18]	Self-Tallying	√	×	×	×
[22]	Self-Tallying	√	√	×	×
[1]	Self-Tallying	√	×	√	√
[20]	Self-Tallying	√	×	×	×
Ours	Voluntary-Tallying	√	√	√	√

not support this feature which is requested for IoT system. We call it as "*voluntary-tallying*". The aforementioned works do not support voluntary-tallying, as well to deal with the abortive or adaptive issues by restarting the election.

3. Preliminaries

3.1. Hardness assumptions

Let $\mathbb{G}$ be a cyclic group of prime order q and g a generator of $\mathbb{G}$.

Definition 1 (*Discrete Logarithm Assumption (DL)*). The Discrete Logarithm assumption is referred to as the following statement: on input g and a random $h \in \mathbb{G}$, there is no probabilistic polynomial time (PPT) algorithm that can compute x such that $g^x = h$ with a non-negligible probability.

Definition 2 (*Decisional Diffie-Hellman Assumption (DDH)*). The Decisional Diffie-Hellman assumption (DDH assumption) is referred to as the following statement: there is no probabilistic polynomial time (PPT) algorithm that can distinguish (g, g^x, g^y, g^{xy}) from (g, g^x, g^y, g^z) where x, y, z are randomly chosen from $\mathbb{Z}_q$.

3.2. Primitives

3.2.1. Linkable ring signature

A linkable ring signature scheme with unconditional anonymity, introduced by Liu, et al. [25], is a ring signature scheme that provides unconditional anonymity but at the same time allows one to determine whether two signatures have been issued by the same group member. A linkable ring signature consists of five algorithms (Setup,KeyGen,Sign,Verify,Link).

- $\mathsf{Setup}(1^\lambda)$: It takes the security parameter λ and outputs system parameters pp.
- $\mathsf{KeyGen}(pp)$: It takes system parameters pp and outputs public/private key pairs $(\mathtt{pk}, \mathtt{sk})$.
- $\mathsf{Sign}(\mathtt{sk}, L, m)$: It takes private key $\mathtt{sk}$, a list L of n public keys and message m. Then, it produces a signature σ.
- $\mathsf{Verify}(L, m, \sigma)$: It takes a list L of n public keys, message m and a signature σ, returns 1 or 0 for accept or reject, respectively.
- $\mathsf{Link}(L, m_1, m_2, \sigma_1, \sigma_2)$: It takes a list L of n public keys, message m_1, m_2 and a signature σ_1, σ_2, returns 1 or 0 for linked or unlinked, respectively. It requires that for any message m_1, m_2, any $L = (\mathtt{pk}_1, \mathtt{pk}_2, \ldots, \mathtt{pk}_n)$ and any $\sigma_1 \leftarrow \mathsf{Sign}(\mathtt{sk}_i, L, m_1), \sigma_2 \leftarrow \mathsf{Sign}(\mathtt{sk}_j, L, m_2)$. If $i = j$, it returns 1. Otherwise, 0.

Review the construction in [25] as follows:

- Setup: Let $\mathbb{G}$ be a group of prime order p that the discrete logarithm problem is intractable. Let $H : \{0,1\}^* \rightarrow \mathbb{G}$ and $H' : \{0,1\}^* \rightarrow \mathbb{Z}_p$ be two hash functions. Let g and h be two generators of $\mathbb{G}$.
- KeyGen: A user randomly chooses $x, y \in \mathbb{Z}_p$ as the private key and computes $Z = g^x h^y$ as the public key.
- Sign: On input $(event, n, \{\mathtt{pk}_i\}_1^n, \mathtt{sk}_\pi, m)$, where *event* is the event description, n is the size of the ring, public key set $\{\mathtt{pk}_1 = Z_1, \ldots, \mathtt{pk}_n = Z_n\}$ of the ring, $\mathtt{sk}_\pi$ is the private key corresponding to the public key $\mathtt{pk}_\pi, \pi \in [1, n]$ and m is the message to be signed, the user computes the following:

–Compute $e = H(event)$ and $\tau = e^x$.
–Randomly generate $r_x, r_y, c_1, \ldots, c_{\pi-1}, c_{\pi+1}, \ldots, c_n \in \mathbb{Z}_p$ and compute

$$K = g^{r_x} h^{r_y} \prod_{i=1,i\neq\pi}^{n} Z_i^{c_i}, \quad K' = e^{r_x} \tau^{\sum_{i=1,i\neq\pi}^{n}}.$$

–Find c_π such that

$$c_1 + \cdots + c_n \mathrm{mod}\ p = H'(\{pk_i\}_1^n || event || \tau || m || K || K').$$

–Compute

$$\hat{x} = r_x - c_x x, \quad \hat{y} = r_y - c_\pi y.$$

–Output the signature $\sigma = (\tau, \hat{x}, \hat{y}, c_1, \ldots, c_n)$.

- Verify: On input $(event, \{\mathtt{pk}_i\}, m, \sigma)$, first compute $e = H(event)$ and

$$c_0 = H'(\{\mathtt{pk}_i\}_1^n || event || \tau || m || g^{\hat{x}} h^{\hat{y}} \prod_{i=1}^{n} Z_i^{c_i} || e^{\hat{x}} \tau^{\sum_{i=1}^{n} c_i})$$

then check whether $\sum_{i=1}^{n} c_i \bmod p = c_0$.

Output `accept` if it is equal. Otherwise, output `reject`.

- Link: On input two signature $\sigma_1 = (\tau_1, \cdot), \sigma_2 = (\tau_2, \cdot)$, two messages m_1, m_2 and an event description *event*, first, check whether two messages are valid. If yes, output `link` if $\tau_1 = \tau_2$ and output `unlink` otherwise.

The LRS with unconditional anonymity requires that for an adversary it should not be possible to identify the identity of the actual signer with a probability significantly greater than $1/n$ when n is the size of the ring.

3.2.2. Non-interactive zero-knowledge proof system [26]

Let R be a relation, corresponding to an NP language L. A non-interactive zero-knowledge (NIZK) proof system for R is a tuple of polynomial-time algorithms $\Pi = (I, P, V)$ specified as follows.

- The randomized algorithm I takes as input the security parameter and outputs a common reference string ω.
- The randomized algorithm $P(\omega, (y, x))$, given $(y, x) \in R$ outputs a proof π.
- The deterministic algorithm $V(\omega, (y, \pi))$, given an instance y and a proof π outputs either 0 (for “reject”) or 1 (for “accept”).

We say that a NIZK for relation R is correct if for all $\lambda \in N$, every ω output by $I(1^\lambda)$, and any $(y, x) \in R$, we have that $V(\omega, (y, P(\omega, (y, x)))) = 1$.

- **Completeness.** If P, V honestly follow the NIZK with $(x, w) \in R, V$ will always accept the proof provided by P.
- **Soundness.** A NIZK Π for a relation R satisfies the knowledge soundness if there exists a PPT extractor $K = (K_0, K_1)$ such that the following holds:
  - Algorithm K_0 outputs ω and an extraction trapdoor ζ, such that the distribution of ω is computationally indistinguishable to that of $I(1^\lambda)$.
  - For all PPT adversaries $\mathcal{A}$, we have that

$$\Pr\begin{bmatrix} V(\omega, y, \pi) = 1 & (\omega, \zeta) \leftarrow K_0(1^\lambda) \\ \wedge (x, y) \notin R: & (y, \pi) \leftarrow \mathcal{A}(\omega) \\ & x \leftarrow K_1(\zeta, y, \pi) \end{bmatrix} \leqslant negl(\lambda).$$

- **Honest Verifier Zero Knowledge (HVZK).** A zero-knowledge proof is a proof that shows the statement is true, but does not reveal anything else. A NIZK Π for a relation R satisfies HVZK if there exists a PPT simulator $Z = (Z_0, Z_1)$ in the condition that all verifiers are honest, such that the following holds:

  - Algorithm Z_0 outputs ω and a simulation trapdoor ζ.
  - For all PPT distinguishers D, we have that

$$|\Pr[D^{P(\omega,\cdot,\cdot)}(\omega) = 1 : \omega \leftarrow I(1^{\lambda}) -$$
$$\Pr[D^{O(\zeta,\cdot,\cdot)}(\omega) = 1 : (\omega, \zeta) \leftarrow Z_0(1^{\lambda})]| \leqslant negl(\lambda),$$

where the oracle $O(\zeta,\cdot,\cdot)$ takes as input a pair (y, x) and returns $Z_1(\zeta, y)$ if $(y, x) \in R$ (and otherwise $\perp$).

We review a concrete non-interactive zero-knowledge proof system as follows [27] in Fig. 2.

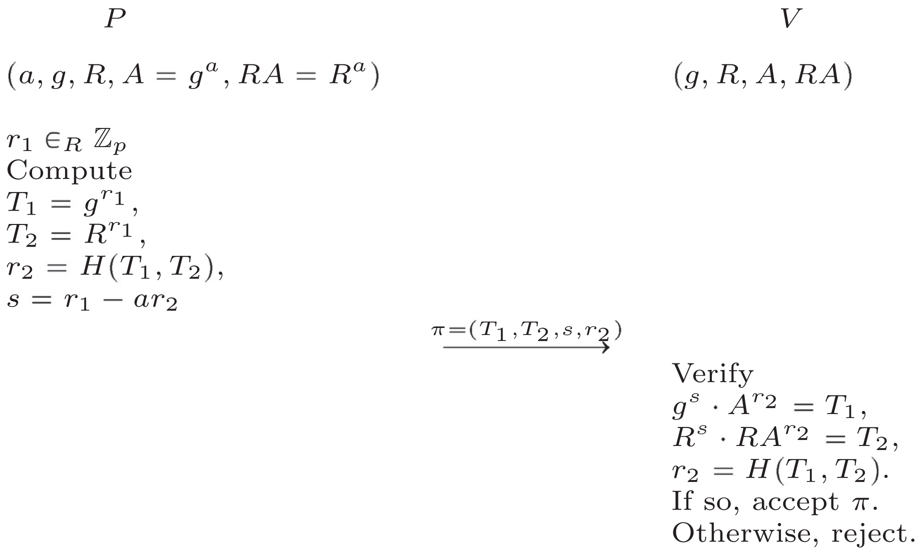
4. System model and security models

The blockchain-based voluntary-tallying leader election for decentralized IoT system is shown in Fig. 3. There are three entities in the system, smart devices, the gateway, and blockchain. In the IoT system, smart devices are as voting devices. The blockchain is able to take over the device management and be a bulletin board. Each device should register in the system when they first enroll then cast the ballot through the gateway to the blockchain. Next, the candidates can check the ballots and compute the auxiliary information and then cast it to the blockchain to claim their ballots, once the ballots have been collected from the blockchain. The tally results for each candidate can be verified by anyone. In this part, we give the system architecture, algorithm definition and security model of our scheme, which contains five phases:

- Setup. It takes the security parameter then outputs the system parameters.
- Registration. It takes the system parameters, then all voters and candidates cast their public/private key pairs.
- Voting. It takes voter's private key and candidate's public key to produce a valid ballot with a weight of 1. Then voters cast the ballot to the blockchain.
- Link. It takes two ballots and check whether they are generated by the same voter. If yes, the second ballot is discarded.
- Tallying. Once the voting phase is over, the candidates check the ballots voting for them and release the auxiliary information that enables anyone who is interest to the tallying result to verify the number of ballots for each candidate.

4.1. Syntax

- Setup(1^{λ}): It takes the security parameter and outputs the system parameter pp. Set the time slots that the registration, voting and tallying are within $time_{reg}, time_{vote}, time_{tally}$, respectively.
- Registration(pp): It takes the system parameter pp and establishes the voter's signing key pair $(\mathtt{pk}_s, \mathtt{sk}_s)$. Each candidate C generates the public/private key pair $(\mathtt{pk}_c, \mathtt{sk}_c)$, during the registration time slot $time_{reg}$.
- Voting$(\mathtt{sk}_s, \mathtt{pk}_c)$: It takes the voter's secret key and the candidate's public key. For voter V_i, it computes the ballot on the elected candidate C_j represented as $ballot$.
- Link$(ballot_1, ballot_2)$: It takes two ballots $ballot_1, ballot_2$. Anyone can determine whether these two ballots are generated by the same voter. If so, the ballots are `link` and the second ballot is discarded. Otherwise, they are `unlink`.



P $\qquad\qquad\qquad\qquad\qquad\qquad$ V

$(a, g, R, A = g^a, RA = R^a)$ $\qquad\qquad$ (g, R, A, RA)

$r_1 \in_R \mathbb{Z}_p$
Compute
$T_1 = g^{r_1}$,
$T_2 = R^{r_1}$,
$r_2 = H(T_1, T_2)$,
$s = r_1 - ar_2$

$$\xrightarrow{\pi = (T_1, T_2, s, r_2)}$$

Verify
$g^s \cdot A^{r_2} = T_1$,
$R^s \cdot RA^{r_2} = T_2$,
$r_2 = H(T_1, T_2)$.
If so, accept π.
Otherwise, reject.

Fig. 2. $PoK\{(a) : A = g^a \wedge RA = R^a\}$.

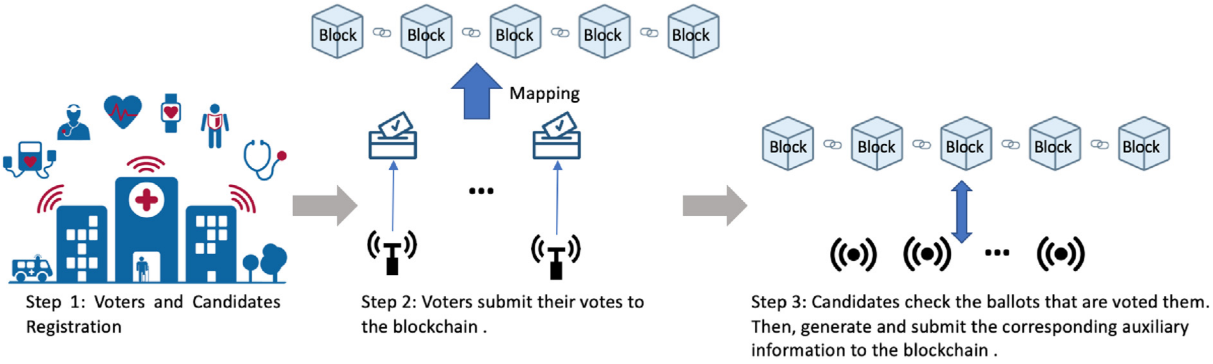

Block
Block
Block
Block
Block
Mapping
Step 1: Voters and Candidates Registration
Step 2: Voters submit their votes to the blockchain .
Block
Block
Block
Block
Block
Step 3: Candidates check the ballots that are voted them. Then, generate and submit the corresponding auxiliary information to the blockchain .

Fig. 3. System Model of the Anonymous Voluntary-tallying Leader Election System.

- $\mathsf{Tallying}(\{ballot_i\}, \mathtt{pk}_c, \mathtt{sk}_c)$: It inputs all the ballots cast in the voting phase and a candidate's public/private key pair. The candidate identifies ballots voted for him/her and compute the auxiliary information I_{aux} for the public tallying. The public can then verify the result for each candidate.

4.2. Security Models

In this section, we formalize the security models for ballot secrecy, voter anonymity and soundness.

Ballot Secrecy (BS). Suppose there are maximal n-2 corrupted voters in the ballot secrecy game, who are fully controlled by the adversary. The adversary can control the corrupted voter's ballot, and also get access to the final result of the leader election. In the challenge phase, given two ballots from two uncorrupted voters, the adversary needs to figure out which of the two ballots is designed to vote the specific candidate. The detailed security model is as follows.

Definition 3 (*Ballot Secrecy*). We say a voluntary-tallying leader election scheme is BS-secure if no polynomial bounded adversary $\mathcal{A}$ has a non-negligible advantage against a challenger $\mathcal{C}$ in the following game:

- **Initial.** There are n voters and $\mathcal{C}$ declares two target voters $\{V_0, V_1\}$ to be challenged. Other voters are corrupted by the adversary.
- **Setup.** $\mathcal{A}$ generates the private and public key pairs for all corrupted voters. The uncorrupted voters and candidates generate their public/private key pairs. Then all voters and candidates publish their public keys.
- **Queries.** $\mathcal{A}$ can control the corrupted voters to generate ballots. $\mathcal{A}$ can make queries on the ballots generated by any voter other than $\{V_0, V_1\}$.
- **Challenge.** $\mathcal{C}$ randomly chooses one voter from $\{V_0, V_1\}$ denoted as V_b. Set the vote of V_b to the candidate C_0 and the vote of V_{1-b} to the candidate C_1 (to simplify the security model, we assume that there are two candidates in the election). $\mathcal{C}$ outputs two challenge ballots on behalf of the uncorrupted voters V_0 and V_1 chosen in the **Initial** phase.
- **Tally.** $\mathcal{A}$ can compute the final result by verifying all auxiliary information with the corresponding ballot.
- **Guess.** $\mathcal{A}$ outputs a guess $b' \in \{0, 1\}$ on b to determine which one between V_0 and V_1 has cast the ballot to C_0. We define the advantage of the adversary wins the game as follows

$$\mathrm{Adv}_{\mathcal{A}}^{\mathrm{BS}}(\lambda) = |\Pr[b' = b] - \frac{1}{2}|.$$

A voting scheme is BS-secure if for any PPT adversary, the advantage is negligible.

Voter Anonymity (VA). Given a ballot generated by one of n voters, by voter anonymity, an adversary should not be able to identify the identity of the actual voter with a probability significantly greater than $\frac{1}{n}$.

Definition 4 (*Voter Anonymity*). We say a voluntary-tallying voting scheme is voter-anonymous, if any adversary $\mathcal{A}$ has a negligible advantage against a challenger $\mathcal{C}$ in the following game.

- **Initial.** $\mathcal{C}$ generates and gives $\mathcal{A}$ the system parameters pp.
- **Queries.** $\mathcal{A}$ queries $\mathcal{C}$ voter's public key, then $\mathcal{C}$ returns the public keys to $\mathcal{A}$.

- **Challenge.** $\mathcal{C}$ chooses a voter V_b from all the voters $\{V_i\}_0^n$ and generates a ballot $ballot_b$ by running the Voting algorithm. $\mathcal{C}$ sends the ballot to $\mathcal{A}$.
- **Guess.** Finally, $\mathcal{A}$ outputs a guess $b' \in \{0, 1, \ldots, n\}$ on $ballot_b$. We define the advantage of the adversary winning the game as follows

$$\mathrm{Adv}_{\mathcal{A}}^{\mathsf{VA}}(\lambda) = |\Pr[b' = b] - \frac{1}{n}|.$$

A voting scheme is voter anonymous if for any PPT adversary, the advantage is negligible.

Soundness. Given a ballot and the auxiliary information from a candidate, any party can check if the ballot is for the candidate. The soundness guarantees that a ballot can only be claimed by the exact candidate encapsulated in the ballot.

Definition 5 (*Soundness*). We say a voluntary-tallying voting scheme is sound, if no polynomial bounded adversary $\mathcal{A}$ can win the following game with a non-negligible advantage.

- **Initial.** $\mathcal{C}$ sets up the system with the security parameter λ.
- **Setup.** $\mathcal{A}$ generates the public/private key pair for all candidates and voters. Then, all public keys will be published to the public.
- **Queries.** $\mathcal{A}$ controls the voters to generate the ballot. Then, $\mathcal{C}$ records a list $List_{vote}$ to be $(C_i, \{\mathtt{ballot}_i\})$ to record the votes on i-th candidate C_i. $\mathcal{A}$ claims the ballots with the candidates' public/private key pairs.
- **Challenge.** $\mathcal{A}$ generates an auxiliary information I'_{aux} to claim a ballot $\mathtt{ballot}' \in \{\mathtt{ballot}_i^*\}$, where $\{\mathtt{ballot}_i^*\}$ is the ballot set for candidate C^* as recorded in the list $List_{vote}$. However, $(I'_{aux}, \mathtt{ballot}')$ is verified successfully for candidate C', which means $\mathtt{ballot}')$ is claimed by C' rather than C^*. Then we say $\mathcal{A}$ $\mathtt{Wins}$ the game. We define the advantage of the adversary winning the game as follows

$$\mathrm{Adv}_{\mathcal{A}}^{\mathsf{Sound}}(\lambda) = \Pr[\mathcal{A}\ \mathtt{Wins}].$$

A voting scheme is sound for any PPT adversary, the advantage is negligible.

5. Privacy-preserving Voluntary-tallying Leader Election

5.1. Framework

To clarify the whole process, we describe our leader election in five phases:

i **Setup phase**: In this phase, the vote initiator publishes the system parameters, candidate's information and time slots. The time slots are for the registration, voting and tallying phases. In the registration time slot, the voter can collect the system information and apply them to obtain their keys. Then, the validated voters can cast their ballots during the voting time. The tally phase starts after the tallying time and, of course, no new ballot can be accepted in this phase.
ii **Registration phase**: In this phase, the voter can obtain the system information such as the candidate public keys, and establish their key pairs.
iii **Voting phase**: In this phase, the valid voter uses its private key and public information of candidate. It firstly computes the ballot for the chosen candidate with the public key of the candidate. Then it publishes the ballot in the blockchain. The Link algorithm will be used to determine whether the ballot can be linked to a previous ballot. If yes, the ballot is discarded in the consensus protocol; otherwise, the ballot is posted in the blockchain.
iv **Tallying phase**: No one can vote during the tallying phase. The candidates identify the ballots that vote for them and generate the corresponding auxiliary information. Then, anyone who is interested to the tally result can access all ballot information and verify the result with the auxiliary information given by candidates.

5.2. Instantiation

We give an instantiation based on the hardness of discrete logarithm problem and decisional Deffie-Hellman problem.

- $\mathsf{Setup}(1^\lambda)$: It inputs the security parameter and outputs the system parameter pp. Common parameters are: q, a prime number; g, h, generators of the group. Set the time slots that the registration, voting and tallying are within $time_{reg}, time_{vote}, time_{tally}$, respectively.

- $\mathsf{Registration}(pp)$: It inputs the system parameter pp and all voters publish their public/private key pair such as $(\mathtt{pk}_s, \mathtt{sk}_s) = (g^x h^y; (x, y))$. For each candidate C_j there is a set of public/private key pairs $(\mathtt{pk}_c; \mathtt{sk}_c) = ((A_j, B_j); (a_j, b_j))$, where $A_j = g^{a_j}, B_j = g^{b_j}$.
- $\mathsf{Voting}(\mathtt{sk}_s, \mathtt{pk}_c)$: It takes the voter's secret key and the candidate's public key. For voter V_i, it computes the ballot on the selected candidate C_j, that $R_i = g^{r_i}, BA_i = A_j^{r_i} \cdot B_j$. Then, the voter treats (R_i, BA_i) as the message to be signed that $(\tau, \hat{x}, \hat{y}, c_1, \ldots, c_n) = \mathsf{LRS.Sig}((R_i, BA_i), \mathtt{sk}_s)$. It finally sends the ballot $ballot = (R_i, BA_i, \tau, \hat{x}, \hat{y}, c_1, \ldots, c_n)$ to the public ballot board.
- $\mathsf{Link}(ballot_1, ballot_2)$: It inputs two ballots $ballot_1, ballot_2$. Any party in the system can determine tags of two ballots whether they are generated by the same voter. If $\tau_1 = \tau_2$, it outputs $\mathtt{link}$. Otherwise, it outputs $\mathtt{unlink}$.
- $\mathsf{Tallying}(\{ballot_i\}, \mathtt{pk}_c, \mathtt{sk}_c)$: It inputs the set of ballots and candidate's public/private key pairs. The candidate computes $RA_i = R^{a_j}$ if $BA_i/RA_i = B_j$, providing RA_i and a proof $\pi = PoK\{a_j : A_j = g^{a_j} \wedge RA_j = R^{a_j}\}$ to the blockchain. In this phase, the candidate only knows the set of members who has voted it, but cannot identify the individual.

  Anyone who is interested to the result can tally it with a set of auxiliary information and ballots, by the following steps. For example, when someone counts the number of votes on C_j, it firstly checks the proof π to ensure that it is acually computed with the secret a_j for C_j. Then, it computes BA_i/RA_i with the auxiliary information RA_i on $ballot_i$ that C_j has claimed. If $BA_i/RA_i = B_i, C_j$ claims $ballot_i$ successfully. It means the number of ballots on C_j adds 1. The verification is anonymous that means the third party has no knowledge of the voters or their ballots.

6. Security analysis

This section is devoted to a theoretical security analysis of our leader election. Our formal security proof is based on the security models given in Section 4.2.

Theorem 1. *There does not exist any PPT adversary who can win the game in ballot secrecy security model with a non-negligible advantage, if the DDH assumption holds.*

Proof. Suppose there are n voters $V_1, \ldots, V_n$ in the game. The challenger $\mathcal{C}$ can interact with the adversary $\mathcal{A}$. We prove Theorem 1 by a sequence of games [28]. We denote $\Pr_i$ as the winning probability of an adversary in **Game i**.

Game 0. This game is the original game defined in Section 3. $\mathcal{A}$ chooses two target voters V_0, V_1 to be challenged and forwards them to $\mathcal{C}$. $\mathcal{C}$ tosses a coin to decide that one of the voters from $\{V_0, V_1\}$ votes the candidate C_0 and the vote of another voter to the candidate C_1 (to simplify the security model, we assume that there are two candidates in the election). By this way, we make $\mathcal{A}$ knows nothing even from the tally result. The one who votes C_0 is denoted by V_b. The challenges consist of $(R_b, BA_b, \mathsf{Sig}_{\mathtt{ssk}}(R_b, BA_b))$. The adversary outputs a guess b' on V_b, then from the definition of the BS game, we have

$$\Pr_0 = \Pr[b = b'] = \Pr_{MBS}.$$

Game 1. This game is the same as **Game 0**. The only difference is to replace the zero-knowledge proof system with a zero-knowledge simulator, and replaces the zero-knowledge proof π with the simulated proofs π' without using the real witness, which is indistinguishable from $\mathcal{A}$'s view. If $\mathcal{A}$ can distinguish between **Game 1** and **Game 0** with a non-negligible advantage, then we can use the adversary to construct an algorithm to break zero-knowledge property of *PoK*. Thus, the adversary's winning probability in Game 1 satisfies the following equation

$$|\Pr_1 - \Pr_0| \leqslant \epsilon_{ZK}.$$

Game 2. This game is the same as **Game 1**. The only difference is to replace the challenged BA_b with a random group element h. $\mathcal{C}$ sets the generator of the group as g. $\mathcal{C}$ selects a voter from $\{V_0, V_1\}$. It sets C_0's public key as (g^x, g^{c_0}). Another uncorrupted candidate C_1's public key is (g^{a_1}, g^{c_1}). $\mathcal{C}$ chooses $b \in \{0, 1\}$ and sets the challenged ballot $ballot_b$ on the chosen voter V_b as $(R_b = g^y, BA_b = h \cdot g^{c_0}$. V_{1-b}'s ballot is $(R_{1-b} = g^{r_{1-b}}, BA_{1-b} = g^{r_{1-b} a_1} g^{c_1})$. $\mathcal{A}$ analyzes which of $\{V_0, V_1\}$ stands for V_b that votes C_0.

This replacement is indistinguishable from $\mathcal{A}$'s view. If $h = g^{xy}$ Game 2 is the same as Game 1. Suppose that $h \neq g^{xy}$ but $\mathcal{A}$ can tell what V_b stands for candidate C_0 with a non-negligible advantage, we can use $\mathcal{A}$ to construct an algorithm to break the DDH instance. Thus, $\mathcal{A}$'s winning probability in **Game 2** satisfies the following equation

$$|\Pr_2 - \Pr_1| \leqslant \epsilon_{DDH}.$$

Game 3. This game is the same as **Game 2**. The only difference is to replace the another voter V_{1-b}'s BA_{1-b} with a random group element $h' \cdot g^{c_1}$. By the same reduction, we can conclude that $\mathcal{A}$'s winning probability in **Game 3** satisfies the following equation

$$|\Pr_3 - \Pr_2| \leqslant \epsilon_{DDH}.$$

Therefore, $\mathcal{A}$ breaks the ballot secrecy security of our voting scheme with the advantage

$$\mathrm{Adv}^{\mathrm{BS}}_{\mathcal{A}}(\lambda) = \Pr_0 \leqslant 2\epsilon_{DDH} + \epsilon_{ZK}.$$

Theorem 2. *There does not exist any adversary who can win the game in the voter anonymity model with a non-negligible advantage, if the linkable ring signature holds the anonymity.*

Proof. Suppose that there are n voters $V_1, \ldots, V_n$ in the game. $\mathcal{C}$ runs Setup to generate the system parameters *pp*. $\mathcal{C}$ runs KeyGen to generate public/private key pairs for all voters and candidates, then it sends the system parameters *pp* and all voters' public/private key pairs $\{(\mathtt{pk}_s^i, \mathtt{sk}_s^i)\}, i \in \{0, \ldots, n\}$ to $\mathcal{A}$. $\mathcal{C}$ sets the public key of the candidate to be challenged as (g^{a^*}, g^{b^*}) and (a^*, b^*) as the corresponding private key.

$\mathcal{C}$ generates the ballots for one of voters in $\{V_i\}_0^n$ by running Voting algorithm. For the ballot $ballot_b$ on candidate C_j, it is in the form of

$$\{R_b, BA_b, \mathsf{LRS.Sign}((R_b, BA_b), \mathtt{sk}_s^b)\},$$

where $R_b = g^{r_b}, BA_b = A_b^{r_b} \cdot B_j$.

$\mathcal{C}$ sends the ballot to $\mathcal{A}$. $\mathcal{A}$ outputs a guess $b' \in \{0, 1, \ldots, n\}$ on b.

Since R_b and BA_b only contain candidate's key and the index of R_b is randomly chosen from the finite field, they are identical from $\mathcal{A}$'s view. Besides, LRS is anonymous. If $\mathcal{A}$ can successfully give a guess on $b, \mathcal{C}$ will construct an algorithm with $\mathcal{A}$ to break the anonymity of LRS scheme. Hence, $\mathcal{A}$ will give a guess b' on b with the advantage

$$\mathrm{Adv}^{\mathrm{VA}}_{\mathcal{A}}(\lambda) = |\Pr[b' = b] - \frac{1}{n}| = 0.$$

Therefore, the self-tallying voting scheme is voter-anonymous.

Theorem 3. *There does not exist any PPT adversary who can win the game in soundness model with a non-negligible advantage, if the underlying zero-knowledge proof is sound.*

Proof. The auxiliary information $I_{aux} = RA_i$ is added to the blockchain with corresponding proof π, generated by using zero-knowledge proof. The soundness of our scheme is guaranteed by the soundness of zero-knowledge proof system.

$\mathcal{C}$ sets up the system. All candidates and voters are controlled by $\mathcal{A}$. For each candidate, $\mathcal{A}$ chooses $a_i \in \mathbb{Z}_p$ to $A_i = g^{a_i}$. Then, $\mathcal{A}$ can simply compute $RA_i = R_i^{a_i}$ corresponding to R_i in the ballot. Also, $\mathcal{A}$ can generate the eligible proof π on this statement (A_i, R_i, RA_i).

$\mathcal{C}$ records the auxiliary information and the corresponding proof on $\mathtt{ballot}_i$ in the list as $\{A_i, R_i, RA_i, \pi_i\}$ and records each candidate's vote in the list $List_{vote}$ as $\{C_i, \{\mathtt{ballot}_i\}\}$.

If $\mathcal{A}$ chooses a ballot $ballot^*$ in the form of $(R^*, BA^*, \mathsf{LRS.Sig}((R^*, BA^*), \mathtt{sk}_s^*))$ on the candidate C^* to be challenge. To produce a proof to prove $ballot^*$ on candidate C', it computes $RA' = BA^*/B'$, where B' is the public key for some other candidate C'. Then, $\mathcal{A}$ has to produce a proof π' on the incorrect statement $(A' = g^{a'}, R^* = g^{r^*}, RA' = g^{r^*a^*+b^*-b'})$ to claim the ballot on C', where A' is the public key of C', where $b^* - b' \neq 0$.

However, the correct proof π^* should be on the statement $(A^* = g^{a^*}, R^* = g^{r^*}, RA^* = g^{r^*a^*})$, where $RA^* = BA^*/B^*$ (B^* is the public key of the challenged candidate C^*). The challenged ballot $\mathtt{ballot}^* \notin \{ballot'\}$ in the list $List_{vote}$ as $(C', \{ballot'\})$. Then, the tallying result will be different from the records in $List_{vote}$. Therefore, we define this case as $\mathcal{A}$ *Wins*.

If $\mathcal{A}$ provides such proof π' successfully. $\mathcal{C}$ can use the incorrect statement $\{A', R^*, RA'\}$ and the proof π' to break the soundness of the zero-knowledge proof system. Hence, $\mathcal{A}$ will win the game with the advantage

$$\mathrm{Adv}^{\mathrm{Sound}}_{\mathcal{A}}(\lambda) = \Pr[\mathcal{A}\ \mathtt{Wins}] \leqslant \epsilon,$$

where ϵ is negligible.

7. Performance analysis

7.1. Experimental analysis

This section discusses the performance of our voluntary-tallying leader election. The analysis is based on the computation time of two processing steps, vote casting performance and tallying performance. All tests were performed using a 512-bit key. We test our proposed scheme on a laptop with the following environmental parameters: 2.8 GHz quad-core Intel Core i7 with 6 MB shared L3 cache and with 16 GB of 1600 MHz DDR3L onboard memory. We test the efficiency of each algorithm when the number of voters increases. The implementation is deployed with Java. The results are illustrated as follows. The time of vote casting is measuring the time cost of Voting for generating a ballot when there are 3, 5, 10 and 20 voters. The time of vote tallying is measuring the time cost of Tallying for generating the auxiliary information and counting the number of ballots for one candidate when there are 3, 5, 10 and 20 voters who have voted it.

For simplifying the condition, here we assume that all voters have voted the same candidate. The ring size of the deployed ring signature is the total number of voters. For simplifying the comparison, we assume that there are only one candidate in the experiments. As we can see from all figures that the running time of Voting and Tallying are $\mathcal{O}(n)$ for voters and the public. For the candidates, the running time of generating the auxiliary information in Tallying phase is $\mathcal{O}(n^2)$.

Vote Casting. According to the Voting, we use Fig. 4a to denote the total time for a voter generating a ballot including the corresponding signature. .

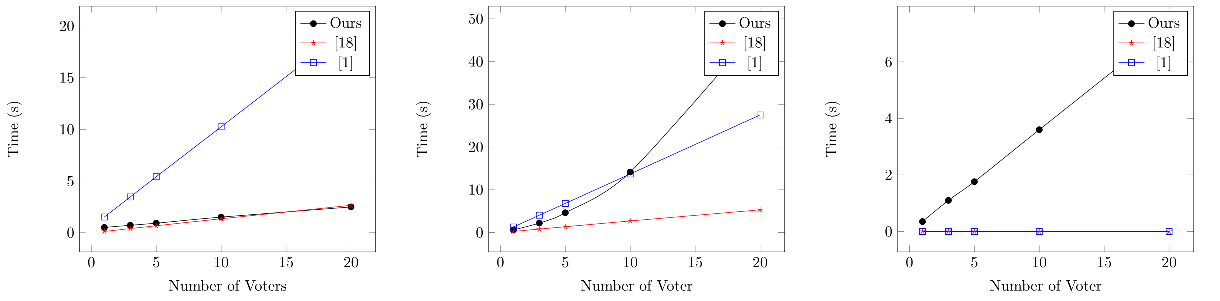
Vote Tallying. Our proposed system allows anyone who is interested to verify all claimed votes. However, before tallying starts, each vote must be verified using the corresponding proofs. Fig. 4b is used to demonstrate the time that candidates spend on signature verification in ballots and generate the auxiliary information. Fig. 4c shows the time that individuals spend on proof verification and count the results.

To avoid the abortive issues and adaptive issues, we take the authorization mechanism in which the valid ballots should be authorized by candidates. By this way, all ballots seems to be randomly, in the voting phase. Any voter can not know how others vote and who generates the ballot. For achieving the voluntary tallying, the candidates also have the rights to claim the ballots on themselves. However, in the conventional self-tallying type schemes, there is no need to provide such functionality to candidates. Thus, in conventional self-tallying type schemes [18,1] the candidates are free from the tallying phase. It is a trade-off between functionality and computation.

7.2. Theoretical comparison

We summarize the computation cost of [18,1], and our proposed scheme. The comparison is over the efficiency and functionality among the above schemes. We assume there are n voters in the systems and c_i indicates the vote on candidate C_i. The voting cost is on generating one ballot and the tallying cost is for one candidate votes counting. The comparison is given in Table 2.

[18] requires voters to generate a zero-knowledge proof for publishing their ballots, then verify other's proofs. In the tally phase, the participants have to provide a zero-knowledge proof to prove the ballot's validation and the public will verify all these proofs while tallying. [1] suggests the participant to give a commitment with a zero-knowledge proof on the committed value. The participants will provide the ballot with a zero-knowledge on it. If the abortive issue occurs, the commitment will be open and verify the corresponding zero-knowledge. In our leader election, we avoid the abortive issue. For compute a ballot, the voter will provide an anonymous ring signature on the ballot. In the tally phase, the candidates verify all the signature and claim the ballots (only the ballots on themselves) with a zero-knowledge proof. The one who is interest to the final result will verify those proofs. Our proposed scheme is the only one that can provide the voter anonymity comparing to the existing schemes. Besides, our scheme does not need repeat the voting phase suppose that there exist some voters



(a) Time of Vote Casting (b) Time Cost of The Candidate during Tallying (c) Time Cost of The Public during Tallying

Fig. 4. Experimental Performance on Voting and Tallying Phases.

Table 2
The Efficiency and Functionality Comparison.

	Voting	Tallying	
		Public	Candidate
[18]	$nM + 3nE$	$2nM + 2nE$	NA
[1]	$(7n+1)M + (9n+15)E$	$(9n-1)M + 16nE$	NA
Ours	$(n+2)M + (n+6)E$	$3nM + 4nE$	$(n^2+3n)M + (n^2+5n+2c_i)E$

who reject to cast their ballots or do not generate their ballots. Although our scheme takes more computation during the tallying phase, our scheme provides voter privacy and fault tolerant without further cost.

8. Conclusion

In this paper, we present a blockchain-based voluntary-tallying leader election system with both ballot privacy and voter anonymity. We give a solution to the adaptive and abortive issues in the conventional self-tallying type schemes by suggesting a novel leader election paradigm, the voluntary-tallying leader election. We also provide the concrete construction and formally prove the security of the construction.

The future work can focus on reducing the communication cost during the tallying phase for the restricted communication and computation capacity of the nodes in some IoT systems. We will pay efforts to implement the novel full-fledged ring signature scheme to achieve more efficient ballot generation algorithm in the voting phase.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work is partially supported by National Natural Science Foundation of China (Grant No. 61872041, U1836212).

References

[1] Y. Li, W. Susilo, G. Yang, Y. Yu, D. Liu, M. Guizani, A blockchain-based self-tallying voting scheme in decentralized iot, CoRR abs/1902.03710. .

[2] C. Xu, J. Wang, L. Zhu, C. Zhang, K. Sharif, PPMR: A privacy-preserving online medical service recommendation scheme in ehealthcare system, IEEE Internet Things J. 6 (3) (2019) 5665–5673.

[3] M. Shen, B. Ma, L. Zhu, X. Du, K. Xu, Secure phrase search for intelligent processing of encrypted data in cloud-based iot, IEEE Internet Things J. 6 (2) (2019) 1998–2008.

[4] M. Li, L. Zhu, X. Lin, Efficient and privacy-preserving carpooling using blockchain-assisted vehicular fog computing, IEEE Internet Things J. 6 (3) (2019) 4573–4584.

[5] C. Zhang, L. Zhu, C. Xu, X. Liu, K. Sharif, Reliable and privacy-preserving truth discovery for mobile crowdsensing systems, IEEE Transactions on Dependable and Secure Computing..

[6] V. Santos, J.P. Barraca, D. Gomes, Secure decentralized iot infrastructure, in: 2017 Wireless Days, Porto, Portugal, March 29–31, 2017, 2017, pp. 173–175..

[7] A. Vangala, B. Bera, S. Saha, A.K. Das, N. Kumar, Y.H. Park, Blockchain-enabled certificate-based authentication for vehicle accident detection and notification in intelligent transportation systems, IEEE Sensors Journal..

[8] S. Saha, D. Chattaraj, B. Bera, A. Kumar Das, Consortium blockchain-enabled access control mechanism in edge computing based generic internet of things environment, Transactions on Emerging Telecommunications Technologies e3995..

[9] B. Bera, S. Saha, A.K. Das, N. Kumar, P. Lorenz, M. Alazab, Blockchain-envisioned secure data delivery and collection scheme for 5g-based iot-enabled internet of drones environment, IEEE Trans. Veh. Technol. 69 (8) (2020) 9097–9111.

[10] B. Bera, D. Chattaraj, A.K. Das, Designing secure blockchain-based access control scheme in iot-enabled internet of drones deployment, Comput. Commun. 153 (2020) 229–249.

[11] S.B. Alla, A. Ezzati, A. Mohsen, Gateway and cluster head election using fuzzy logic in heterogeneous wireless sensor networks, in: 2012 International Conference on Multimedia Computing and Systems, IEEE, 2012, pp. 761–766.

[12] A. Wood, G. Virone, T. Doan, Q. Cao, L. Selavo, Y. Wu, L. Fang, Z. He, S. Lin, J. Stankovic, Alarm-net: Wireless sensor networks for assisted-living and residential monitoring, University of Virginia Computer Science Department Technical Report 2 (2006) 17.

[13] D. Chaum, Untraceable electronic mail, return addresses, and digital pseudonyms, Commun. ACM 24 (2) (1981) 84–88.

[14] K. Sako, J. Kilian, Secure voting using partially compatible homomorphisms, in: Advances in Cryptology - CRYPTO '94, 14th Annual International Cryptology Conference, Santa Barbara, California, USA, August 21–25, 1994, Proceedings, 1994, pp. 411–424..

[15] J.D. Cohen, M.J. Fischer, A robust and verifiable cryptographically secure election scheme (extended abstract), in: 26th Annual Symposium on Foundations of Computer Science, Portland, Oregon, USA, 21-23 October 1985, 1985, pp. 372–382. .

[16] A. Kiayias, M. Yung, Self-tallying elections and perfect ballot secrecy, in: Public Key Cryptography, 5th International Workshop on Practice and Theory in Public Key Cryptosystems, PKC 2002, Paris, France, February 12–14, 2002, Proceedings, 2002, pp. 141–158..

[17] F. Hao, P. Zielinski, A 2-round anonymous veto protocol, in: Security Protocols, 14th International Workshop, Cambridge, UK, March 27–29, 2006, Revised Selected Papers, 2006, pp. 202–211..

[18] F. Hao, P.Y.A. Ryan, P. Zielinski, Anonymous voting by two-round public discussion, IET Inf. Secur. 4 (2) (2010) 62–67.

[19] P. McCorry, S.F. Shahandashti, F. Hao, A smart contract for boardroom voting with maximum voter privacy, in: Financial Cryptography and Data Security - 21st International Conference, FC 2017, Sliema, Malta, April 3–7, 2017, Revised Selected Papers, 2017, pp. 357–375..

[20] X. Yang, X. Yi, S. Nepal, F. Han, Decentralized voting: A self-tallying voting system using a smart contract on the ethereum blockchain, in: Web Information Systems Engineering - WISE 2018–19th International Conference, Dubai, United Arab Emirates, November 12–15, 2018, Proceedings, Part I, 2018, pp. 18–35..

[21] J.J. Kilian, K. Sako, Secure anonymous message transfer and voting scheme (1997)..

[22] J. Groth, Efficient maximal privacy in boardroom voting and anonymous broadcast, in: Financial Cryptography, 8th International Conference, FC 2004, Key West, FL, USA, February 9–12, 2004. Revised Papers, 2004, pp. 90–104..

[23] P. Aradhya, Distributed ledger visible to all? ready for blockchain, Huffington Post..

[24] B. Wire, Now you can vote online with a selfie. business wire (2016)..

[25] J.K. Liu, M.H. Au, W. Susilo, J. Zhou, Linkable ring signature with unconditional anonymity, IEEE Trans. Knowl. Data Eng. 26 (1) (2014) 157–165.

[26] C. Schnorr, Efficient signature generation by smart cards, J. Cryptology 4 (3) (1991) 161–174.

[27] D. Chaum, T.P. Pedersen, Wallet databases with observers, in: Advances in Cryptology - CRYPTO '92, 12th Annual International Cryptology Conference, Santa Barbara, California, USA, August 16–20, 1992, Proceedings, 1992, pp. 89–105..

[28] V. Shoup, Sequences of games: a tool for taming complexity in security proofs, IACR Cryptology ePrint Archive 2004 (2004) 332.